Marketing Decision Support Systems: A Role for Neural Networking

Proctor, R. A.. **Marketing Intelligence & Planning**. Bradford: 1992. Vol. 10, Iss. 1; pg. 21, 6 pgs

» Jump to full text

Subjects:	Statistical analysis, Software reviews, **Neural** networks, Marketing information systems, Decision s
Classification Codes	9120 Product specific treatment, 7000 Marketing, 5240 Software & systems, 2600 Management sci
Companies:	**Ward Systems Group** Inc
Author(s):	Proctor, R. A.
Publication title:	Marketing Intelligence & Planning. Bradford: 1992. Vol. 10, Iss. 1; pg. 21, 6 pgs
Source Type:	Periodical
ISSN/ISBN:	02634503
ProQuest document ID:	8948311
Text Word Count	2465
Article URL:	http://gateway.proquest.com/openurl?ctx_ver=z39.88-2003&res_id=xri:pqd&rft_val_fmt=ori:fmt:kev:mtx:journal&genre=article&rft_id=xri:pqd:did=000000008

More Like This »Show Options for finding similar articles

Abstract (Article Summary)

Neural network technology is an alternative expert system for solving marketing decision problems. **Neural** network technology can be used to produce a logic system that is not intended to give exact solutions to problems, but to give reasonable estimations. One of the most sturdy types of **neural** network **algorithms** is the back propagation **algorithm**. Neuroshell, from **Ward Systems Group** Inc., is a **neural** network shell that uses this **algorithm**. Neuroshell may seem to offer little advantage over standard regression analysis. However, this method is very simple to use and offers considerable advantage to those not skilled in statistics. Neuroshell works best where there is a high R squared value present in the statistical analysis. The fact that Neuroshell cannot learn all data patterns is not a particular disadvantage, since standard statistical methods will indicate that the data will have a poor fit to the model in cases where Neuroshell is unable to learn the data.

Full Text (2465 words)

MARKETING INFORMATION SYSTEMS--A GROWING FUNCTION IN A BUSINESS

In 1989 Martech (1989) produced a survey into marketing information systems in the UK. Its conclusion was that there did appear to be a general recognition of the benefits of computerization within marketing and stated that there were trends in the marketplace that have made information management more critical to the achievement of competitive advantage. It was indicated that computerization of marketing has grown very rapidly over the past two years with penetration increasing from 15 per cent to about 50 per cent. Growth in applications has been concentrated on fragmented solutions to tackle a specific element of the sales/marketing function, particularly in salesforce productivity. Very few companies have moved beyond the isolated productivity tool stage towards integrated marketing information systems.

There is a large role for computers to play as part of the marketing decision support system. Databases,

spreadsheets, expert systems and other applications packages can play an important part in this process. This article considers the contribution that a particular expert system can play. The system is user-friendly and has the advantage over other types of expert systems in that it is extremely easy to use.

NEURAL NETWORKING--AN OVERVIEW

Neural network technology can be used to produce a fuzzy logic system that is not intended to give exact solutions to every problem but will give reasonable estimations. A fuzzy logic system (FLS) classifies patterns according to other patterns that it has "learned" and on the basis of this learning it suggests solutions. If patterns are incomplete or in some way inharmonious, solutions produced may not always be very precise.

There are various types of **neural** network algorithms that vary in proficiency and suitability. One of the most sturdy types is the "back propagation" algorithm and it is used for many kinds of problems. This article looks at one new network shell that uses the back propagation algorithm and examines how it might be used to solve problems in marketing.

The chief constituent of **neural** network techniques is the simulated neuron which processes several inputs to produce an output. Both inputs and outputs take on numeric values between O and 1. Values close to 1 represent positive stimulation whereas values close to O imply negative or no stimulation. For practical purposes this means that when presenting data sets to **neural** network devices the data values have first to be standardized between O and 1. Neurons may be linked to one another so that the output from one or more neurons may become the inputs for other neurons. Neurons may also make inputs and create outputs to the outside world. A neuron creates an output by summing the inputs and then applying a linear function to the sum. This is called an output function.

Neurons need to be interconnected in a network of neurons, which are called nodes. Nodes occur in layers such that there is one input layer of nodes, one output layer of nodes and several hidden layers of nodes that act as mediators between the input and output layers. Each of the input nodes passes its output to each of the output nodes via the hidden nodes. As the first layer passes its values to the second layer the values are weighted to depict connection strengths. To reinforce a connection positively, the weight is lowered. The network processes data b4' accepting input patterns (values between O and 1) into the input nodes and the network produces related output patterns.

Feedback is obtained when patterns to be learned are presented to the input nodes. It establishes whether the reinforcement is positive or negative. Every input pattern has a known output pattern that the network is expected to produce. The error between the expected and actual outputs for a specific output node is calculated. The weights leading to this output node are then modified in the direction required to produce a smaller error when the pattern is resubmitted. Learning occurs as repeated submission of the patterns and weight adjustment takes place. It continues until the total error of all output nodes falls below a predetermined threshold value.

Once learning has been completed, the network should be able to create the correct output pattern when given an input pattern it has learned. Moreover, it can also generalize by identifying an input pattern close to a pattern it has learned and generate an output close to the pattern it has been trained to invoke.

POTENTIAL APPLICATIONS IN MARKETING

Neural networks are able to learn any pattern, however, in this article we will concentrate on considering numerical and verbal data patterns.

In marketing we encounter numerical arrays of different dimensions and have to find some way of reducing the data such that we can readily interpret it. Largely this takes us into statistical analysis of one form or another. By way of example we may wish to make a sales forecast for a product. Let us assume that the forecast to be made is sales of umbrellas and that we also have available other data that will help us in our forecasting. In the illustration, relative price is the price of our umbrellas vis-a-vis the average price charged by our competitors. Relative promotional expenditure is our expenditure on promotion vis-a-vis the average spent by our competitors. Here we might forecast sales of umbrellas on the basis of our expectations regarding relative price, personal disposable income, relative promotional expenditure and rainfall (Table I). (Table I omitted)

A standard statistical approach to this kind of forecasting problem is to use multiple regression analysis. There are many different statistical packages around to help us to do this and perhaps the easiest to use is the "regression" feature in Lotus 123. Applying this latter package feature to the data above produces the following output in Lotus

123:

Regression output:

Constant -- -1402.04 Std. Err of Y Est. -- 101.6809 R squared -- 0.963950 No. of Observations -- 16 Degrees of freedom -- 11

X coefficient(s) -- -569.185 0.056718 1038.137 12.00515 Standard Error of Coefficient -- 414.6614 0.004944 294.9570 8.985038

We can recast the above output into a more understandable form to help us make forecasts. This takes the form of an equation where the equation values are taken from the above.

$Y = k + X1 + X2 + X3 + X4 + u$

where:

Y is sales of umbrellas k is the constant X1 is the relative price X2 is the personal disposable income X3 is the relative amount spent on promotion X4 is the rainfall u is the error in the forecast

which gives:

$Y = -1402.04 - 569.185X1 + 0.056718X2 + 1038.137X3 + 12.00515X4$

Hence to predict sales in any one year one simply enters expected values for X1, X2, X3 and X4. The other statistics in the Lotus 123 printout enable us to put confidence limits on our estimates.

A good example of a **neural** networking tool is Neuroshell. If we give the above data to Neuroshell to "learn" then it will subsequently allow us to enter the relevant data, shown below, and will then forecast sales for us. For example: (data omitted)

Since the forecasts made with Neuroshell above have used the same values as are included in the "past data" tables, we can in fact check the accuracy of these forecasts. Case 1 in fact is the data for 1975, Case 2 is the data for 1976. The accuracy in this example is quite good. This in fact is not too surprising, however, since in the regression analysis we obtained a multiple R2 value of more than 0.96 suggesting that a linear model would fit the data well.

In tackling problems of this kind, where there are several independent variables and a single dependent variable, Neuroshell may seem to offer little advantage over standard regression analysis. Indeed, purists would argue that it is somewhat limited because it does not enable confidence limits to be set for estimates. On the other hand, this method is very simple to use and offers considerable advantage to the person who is not skilled in statistics. However, Neuroshell does work best where there is a high R sup 2 value present in the statistical analysis.

The above example makes use of synthetic data. I have tried out Neuroshell on real data and produced some interesting results. In one case I was trying to relate the prices charged in Europe in 1989 for new cars to various attributes of product quality. I obtained a database of information on characteristics of cars through a shareware organization. The database was stored in dBase3 files. Some of the prices were in Swiss francs whilst others were in German marks. I extracted from the database some 567 cases where the price data was in Swiss francs. Next, using the dBase conversion utility in Lotus 123, I converted the dBase file into a Lotus Spreadsheet file that could then be loaded into Neuroshell.

After subjecting Neuroshell to "learning" the data Neuroshell was able to tell me what price I should expect to charge for a car having various features. An actual example is shown below: (example omitted)

There are obviously many opportunities in marketing for this kind of analysis to be made. Neuroshell is able to produce an expert system that can be used by even the least statistically knowledgeable executive.

HOW **NEURAL** NETWORKS HELP TO SOLVE PROBLEMS WHERE MORE CONVENTIONAL APPROACHES

ARE DIFFICULT TO APPLY

There are some kinds of numerical problems where we are not just trying to satisfy one specific objective. Let us consider the problem where we are trying to achieve three different objectives. When we are screening and evaluating new product opportunities we might draw up a checklist against which to rate ideas. Such a checklist could include items such as:

* fit of the new product with existing pricing policies

* fit with other products in the product mix or line

* fit with promotional expertise

* fit with current distribution policy.

There may be 20 or more items on a checklist. We could then rate product ideas on a nine-point scale against every item on the checklist and sum the scores for every product rated. Products scoring more than a given number might then be moved on to the next stage of the screening process.

An interesting point here is that it would be useful if we could link scores obtained against items on a checklist with the subsequent performance of the product in the marketplace. Suppose we could, from historical records, put together the data shown below (Table II). (Table II omitted) Each row entry represents a product that a firm has launched in the past. The first four columns show the ratings that the firm gave to the product for fit with distribution, pricing, product mix and promotion. The fourth, fifth and sixth columns show the return on investment, months taken to achieve breakeven and sales volume achieved by the products.

In such a case conventional multiple regression analysis fails us for it can only deal with one dependent variable and not three as we have here. Statistical analysis can help through the medium of canonical correlation. However, it is onerous to interpret the results of such an analysis and the methodology does not lend itself readily to making predictions.

Neuroshell, on the other hand, is quite capable of undertaking the required analysis. It can handle multiple dependent variables and multiple independent variables.

The results of applying Neuroshell to the above data are shown below. Again we are looking at the original data and making predictions for the dependent variables based on the input variables. It will be noted that the error in fact is quite small. (results omitted)

NON-NUMERICAL APPLICATIONS FOR NEUROSHELL

Essentially, Neuroshell is an expert system shell that can operate in a binary or analogue mode. So far we have examined its use only in the analogue (numerical) mode. The binary mode enables one to set up an expert system which uses essentially non-numerical data. Kotler (1988) provides an illustration of how a firm might respond to a competitor cutting its price. We can set the various conditions and actions into a decision table (Table III). (Table III omitted)

The top half of Table III shows the various conditions that can occur. A Y indicates the presence of the condition whereas an N or a blank space indicates the condition's absence. In the bottom half of the table the reactions to the conditions are shown; a 1 in the appropriate column signifying the appropriate action to be taken. It will be noted that there are seven possible reactions to take.

Neuroshell handles binary data of this kind as it does analogue or numerical data. In this instance there are seven cases to enter. The cases are learned in the same way as the numerical ones discussed above and having learned the data Neuroshell can then be used to indicate outcomes of various combinations of inputs. The following illustrates its use with the above data (Note: the conditions are shown in bold type, as is the recommended course of action): (example omitted)

It will be noted that the recommended action also has a number of 0.90 associated with it. This number is higher than any of the other possible actions and confirms the aptness of the indicated line of action.

APPRAISAL AND LIMITATIONS OF NEUROSHELL

The key advantage of using such a package in decision support systems is the ease with which it can be used. Even the least numerate person can use the package and it also provides an excellent shell for designing simple or complex expert systems.

Some of the disadvantages we have considered already --for example, the difficulty of establishing confidence limits for predictions. However, for practical use, this is a minor point. Neuroshell also cannot learn all data

patterns. This is not a particular disadvantage since standard statistical methods will indicate that the data will have a poor fit to the model in cases where Neuroshell is unable to learn the data. Moreover, it is unlikely that other approaches would produce better results.

A major criticism of **neural** networks relates to their learning speeds. It can take hours or several days for a machine to learn a particular dataset. The way to solve this problem is to use a high speed personal computer preferably with a maths co-processor in place.

REFERENCES

Kotler, P. (1988), Marketing Management: Analysis, Planning and Control, 6th Edition, Prentice-Hall, Englewood Cliffs, NJ, p. 525.

Martech Information Systems (1989), West Africa House, Ashbourne Road, London, W5 3QR.

NeuroShell, Ward Systems Group Inc., 228 West Patrick Street, Maryland, USA.

FURTHER READING

Aleksander, I. and Morton, H. (1990), **Neural** Computing, Chapman and Hall, London.

Anderson, J.A. and Rosenfeld, E. (1989), Neurocomputing, MIT Press.